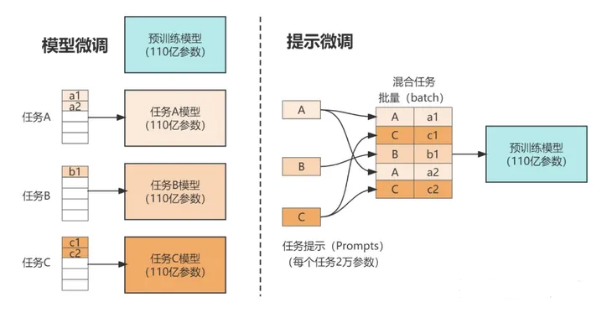
从头开始训练一个深度神经网络可能需要大量的时间和计算资源，并且存在准确率低，容易过拟合等风险。预训练模型是在大规模的通用数据上进行训练的，学到了丰富的特征表示，因此可以节省这些资源。其次在很多实际应用中，获取大规模标注数据是昂贵且耗时的，而预训练模型可以利用通用数据进行训练，然后通过微调在相对较小的标注数据上进行特定任务的训练。为增强模型的泛化能力和迁移能力，我们选择在基于Self-Attention的时间序列预测上使用预训练及高效微调策略。

预训练包括特征提取和微调模型两种方式，为了提高模型的迁移能力，我们选择P-Tuning v2并结合使用预训练模型参数约束P-tuning v2过程的方法进行微调。P-Tuning v2 是一个强大的微调方法，目标是要让 Prompt Tuning 能够在不同参数规模的预训练模型、针对不同下游任务的结果上都达到匹敌 Fine-tuning 的结果。

（1）P-tuning v2

Prompt tuning:固定预训练参数，为每一个任务额外添加一个或多个 embedding层，之后拼接 query 正常输入 LLM，并只训练这些 embedding。左图为单任务全参数微调，右图为 Prompt tuning。



P-tuning v2 方法在prompt tuning上有了以下两方面的改进。

①提出将 Prompt 转换为可以学习的 Embedding 层,使用MLP+LSTM的方式来对 prompt embedding 进行一层处理，具体情况如下：



固定LLM参数，利用多层感知机和LSTM对Prompt进行编码，编码之后与其他向量进行拼接之后正常输入 LLM。

②在多层加入了Prompts tokens作为输入，其中Prompt的长度和数量是影响最后效果的有效参数。

1. 使用预训练模型约束P-tuning v2过程

为了防止模型在P-tuning v2阶段过拟合目标任务数据，或忘记了预训练阶段学到的有意义知识，我们使用预训练模型参数约束P-tuning v2过程的方法：通过添加预训练模型参数和P-tuning v2模型参数之间的L2正则化损失作为微调过程中损失函数的一部分，公式如下（w为P-tuning v2过程参数，w0位预训练参数）：

